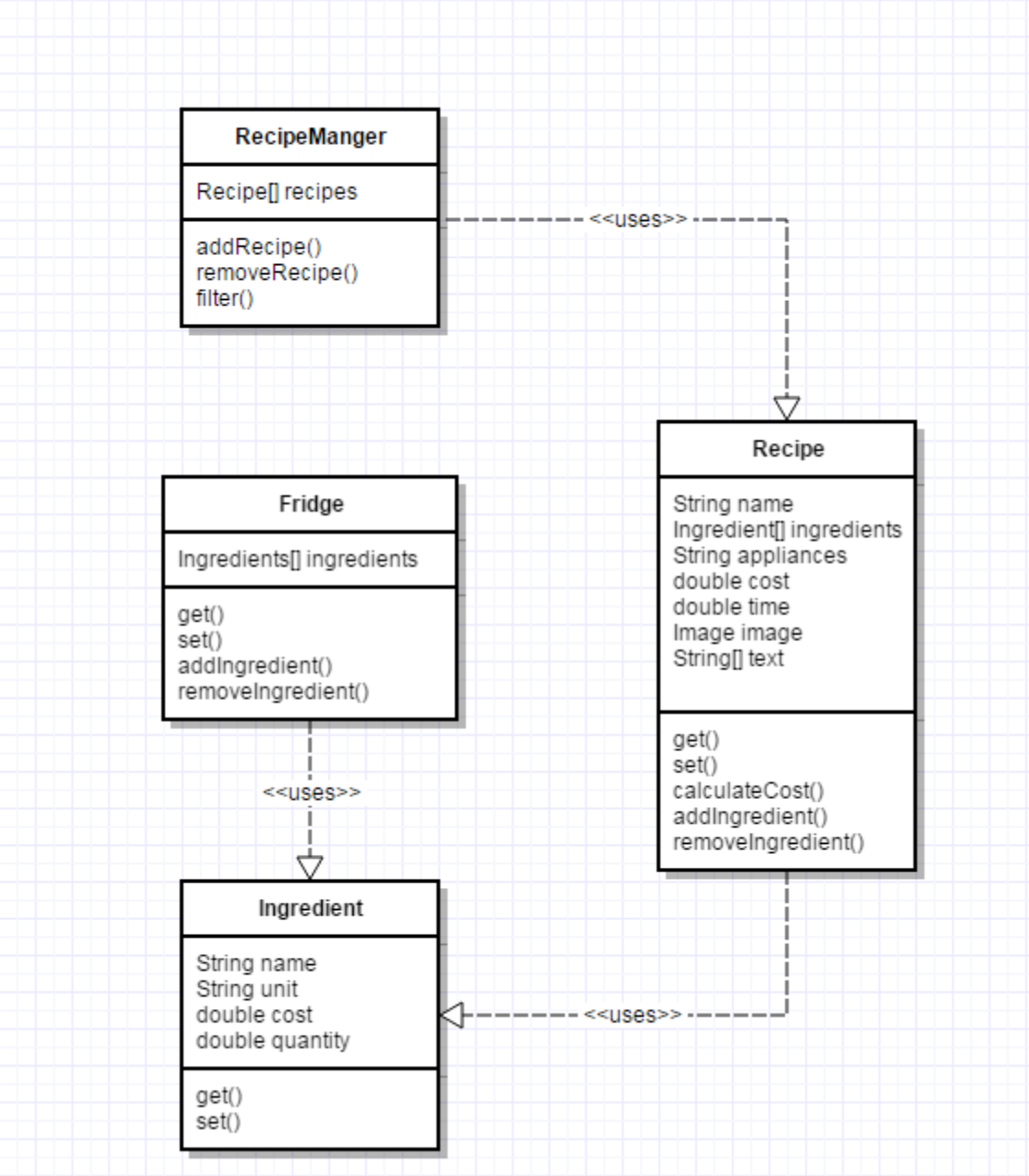
Design Document

Section I - Problem Analysis & Purpose

A typical everyday struggle is the need for food. Eating at home is preferable, but it is often hard to be know what dinners can be made from the ingredients present, or with limited appliances. This leads to dining out- mayhem for a students' wallet.

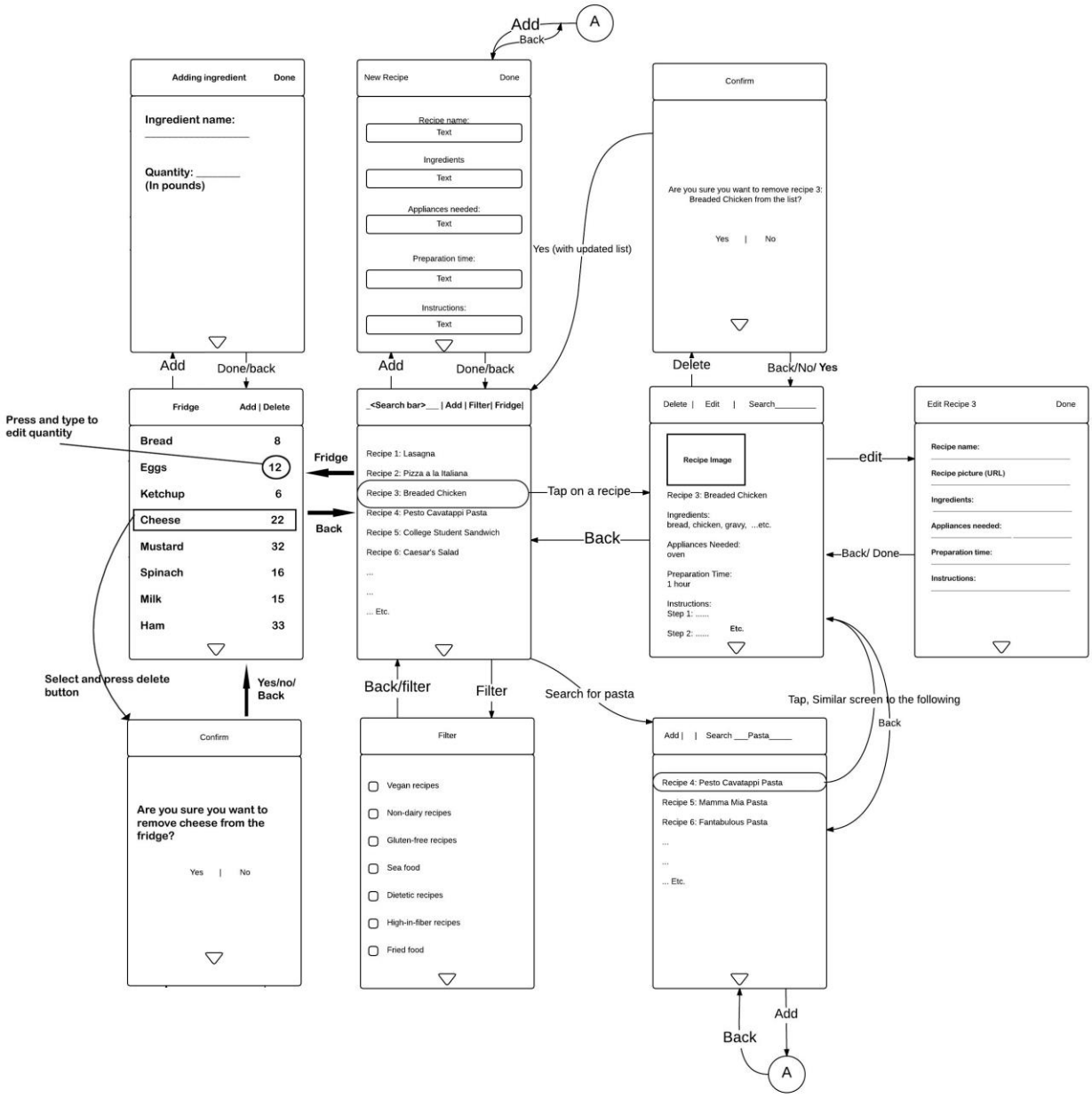
This Android app resolves this dilemma, by providing a user-friendly touch-based interface to enter ingredients into and then traverse a database of recipes containing a variety of meals to make. The app is not limited to simply viewing a recipe, users will be able to save favorites, create their own recipes, filter searches and examine overall costs of meals. Additionally, the user will be able to keep track of the items they currently have in their fridge, and filter recipes with those saved ingredients.

Section II - High-level Entities



- The Ingredient object represents an ingredient used in the Fridge or a Recipe.
- The Recipe object represents all elements of a recipe.
- The Fridge object represents all ingredients in the user's fridge, determined by user input.
- The RecipeManager manages which recipes appear in the app.

The following illustration shows a rough transition diagram between different user-interface components in our app. The menu state located in the middle-left in the diagram corresponds to the home screen of the app.



In this app, the user will be able to:

- Search for menu recipes from our database by either typing a recipe name (or part of its name), scrolling through all the recipes, or using filters and tapping on the desired recipe
- Create, edit and delete recipes

Section III - Low-level Entities

Example XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<recipe>
    <recipename>Scrambled Eggs</recipename>
    <ingredient>
        <name>Egg</name>
        <unit>Egg</unit>
        <costperunit>0.15</costperunit>
        <quantity>2</quantity>
    </ingredient>
    <ingredient>
        <name>Whole Milk</name>
        <unit>cup</unit>
        <costperunit>1.00</costperunit>
        <quantity>0.125</quantity>
    </ingredient>
    <ingredient>
        <name>Butter</name>
        <unit>tbsp</unit>
        <costperunit>0.09</costperunit>
        <quantity>0.5</quantity>
    </ingredient>
    <ingredient>
        <name>Salt</name>
        <unit>teaspoon</unit>
        <costperunit>0.01</costperunit>
        <quantity>0.5</quantity>
    </ingredient>
    <ingredient>
        <name>Pepper</name>
        <unit>teaspoon</unit>
        <costperunit>0.01</costperunit>
        <quantity>0.5</quantity>
    </ingredient>
    <appliance>Stove</appliance>
    <time>15</time>
    <cost>0.48</cost>
    <image>http://tinyurl.com/q6or4u4</image>
    <text>
        <step>Beat together all ingredients.</step>
        <step>Put butter in pan. Bring to medium heat until butter is melted. </step>
        <step>Add egg mixture, and stir gently until done.</step>
```

```
    </text>
</recipe>
```

- The Ingredient object will represent an ingredient used in either a Recipe or a Fridge. The object will contain a name, a unit of measurement appropriate to that object, a cost per unit, and a quantity, in terms of the defined unit.
- The Recipe object represents a recipe, displayed in the app. It contains a list of Ingredients, a name, all used appliances, the preparation time, the aggregate cost of all Ingredients, a representative image, and a list of steps.
- The Fridge object will contain an array of Ingredients, and represents the Ingredients available to the user. The user will be able to add or remove ingredients from the Fridge.
- The RecipeManager will contain a list of Recipes to manage, with functionality to add or remove a recipe, and to filter.

Section IV - Benefits, Assumptions, and Risks

Benefits

- The filter function orders all results based on how many ingredients are matched to each recipe.

Assumptions

- We are assuming a small number of recipes. This simple implementation will likely not be able to support large volumes of data.
- We are assuming security will not be an issue.

Risks

- Storing the data locally will be a risk, due to limits on the mobile memory.